A Realistic Scooter Rebalancing System via Metaheuristics

Guilherme Fernandes
ALGORITMI Center, University of Minho, Portugal

Nuno Oliveira
GoWithFlow, Matosinhos, Portugal

Paulo Cortez
ALGORITMI Center, University of Minho, Portugal
pcortez@dsi.uminho.pt

Rui Mendes
ALGORITMI Center, University of Minho, Portugal
azuki@di.uminho.pt

ABSTRACT

This paper addresses a realistic electric scooter rebalancing task that includes business rules (e.g., time constraints, available truck fleet). We explore an integer encoding approach and three metaheuristics (hill climbing, simulated annealing, genetic algorithm), discussing the obtained results, current limitations and future work directions.

CCS CONCEPTS

• **Computing methodologies → Genetic algorithms**; • **Theory of computation → Simulated annealing**; • **Applied computing →** *Multi-criterion optimization and decision-making*;

KEYWORDS

Bike Sharing Rebalancing Problem, Metaheuristics

ACM Reference Format:
Guilherme Fernandes, Nuno Oliveira, Paulo Cortez, and Rui Mendes. 2020. A Realistic Scooter Rebalancing System via Metaheuristics. In *Genetic and Evolutionary Computation Conference Companion (GECCO '20 Companion), July 8–12, 2020, Cancún, Mexico.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3377929.3389905

1 INTRODUCTION

Many cities are nowadays trying to find smart mobility solutions (e.g., cost efficient and environmental friendly). This study focuses on the bike/vehicle sharing rebalancing problem [3, 4]. Several studies have proposed metaheuristics to solve this task, such as simulated annealing [1] or particle swarm algorithm [5]. However, these studies usually simplify the problem. For instance, they often ignore business and resource constrains (e.g., number of trucks that redistribute the bikes/vehicles and their capacity).

Our approach uses the Barcelona city electric scooter sharing system. Since this system is widely used, it is necessary to reallocate scooters several times a day (the rebalancing task). A realistic reallocation should consider several trucks and their individual capacity which can be distinct, travel time and the operational costs (e.g., the time spent unloading or loading scooters from the truck). Furthermore, it should consider business constraints, since all stations should be balanced after a given amount of time. This paper describes an initial attempt to realistically model this task using metaheuristics, showing obtained results, current limitations and launching future research directions.

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

GECCO '20 Companion, July 8–12, 2020, Cancún, Mexico

ACM ISBN 978-1-4503-7127-8/20/07…$15.00
https://doi.org/10.1145/3377929.3389905

2 THE PROBLEM

The Barcelona electric scooter service contains 264 areas whose demand changes during the day. Our data was collected three times (at 4 am, 10 am and 4 pm) a day during 222 days (664 records). Normally, each record contains 9 to 14 stations with rebalancing needs. Each station i has a relocation need of R_i, where $R_i < 0$ means it needs more scooters and $R_i > 0$ means it has a surplus. Trucks leave the depot and must take all scooters from the stations with a surplus and supply the stations with a deficit. These constraints must be satisfied in order to fulfill the business model:

- each truck t starts at the depot either vacant or with some initial load (L_t) and finally returns to the depot (one truck trip);
- stations should be resupplied as soon as possible and there is a time limit for fulfilling all requests;
- a station may be visited multiple times and be used for temporary storage in order to improve the solution;
- the truck capacity cannot be exceeded and speed limits cannot be broken;
- to reduce costs, the number of trucks and total distance should be minimized.

Our approach assumes several truck trips during the required time window of W minutes: 1) if there are station needs, the metaheuristic is run to set a trip for one truck; 2) if there are further station needs and more trucks, the algorithm is rerun with another truck and step 2) is repeated; 3) if there are more needs, all trucks were used and there is enough time left, go to step 1). The goal is to minimize the distance required to satisfy as many stations as possible, trying to reduce the number of stations left to visit in the next trips. Equation 1 shows the objective function used in this study, where: $d(i, i+1)$ represents the distance between two consecutive stations in the solution; and n is the number of stations that have a balancing need. Supplied stations, when $R_i = 0$, should no longer be considered on the next trips. Finally, α and β denote parameters that weight the distinct subgoals (unsupplied needs and supplied stations) into a single measure:

$$\underset{x}{\text{minimize}} \sum_{i=1}^{n-1} d(i, i+1) + \alpha \cdot \sum_{i=1}^{n} |R_i| - \beta \cdot \sum_{i=1}^{n} [R_i = 0] \qquad (1)$$

3 EXPERIMENTS WITH METAHEURISTICS

We used two trucks ($t \in \{1, 2\}$) capable of carrying 15 scooters with a time window of $W = 60$ minutes for fulfilling all demands (from a specific instance of the problem, with n =10 stations) and with a load time of one minute per scooter. The experiments were conducted with the R tool using Hill Climbing (HC), Simulated Annealing (SA) and a Genetic Algorithm (GA), via HC code from [2] and SANN from the optim package ($T = 10$, $maxIt = 10,000$, logarithmic cooling schedule) and GA packages. To reduce the computational effort, all experiments were executed using only 10 runs. The encoding chosen (Figure 1) uses integer values for the visited stations (S_i) and truck loads (L_i, leaving the station), thus setting a truck trip (which always starts and ends on depot). Values were constrained in order to maintain consistency, making sure that it is not possible to have invalid stations or cargoes. New solutions are generated by using mutation (HC, SA and GA) or one-point crossover (GA). Figure 2 shows the mutation operators for updating the stations. The differential mutation operator was used for changing cargo loads, which adds or removes 1 from a load (but enforcing the cargo constraints). The GA was set with a population of 50 individuals, linear rank selection with 5% elitism, a crossover probability of 0.4 and a mutation probability of 0.6 (with equal chances of choosing any operator). Also, there is a solution repair mechanism that prevents station duplication or travels between stations without cargo change.

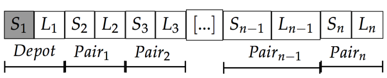

Figure 1: Adopted solution representation

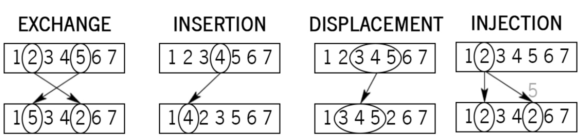

Figure 2: Mutation operators for modifying stations

After preliminary experiments (using grid search), the objective function parameters were set to $\alpha = 1500$ and $\beta = 100$. We also used a custom initialization, where stations were chosen depending on whether the truck had enough scooters to satisfy their needs (a station with deficit is chosen) or if the truck is almost empty (a station with excess is chosen). The last station before the depot was always a station with a deficit in order for the truck to return to the depot as empty as possible. One fifth of the population was created with random stations while enforcing the constraints, two fifths were chosen rationally but with random loads, while the last two fifths chose rational stations but with maximum loads (by loading all the surplus scooters or supplying as much of the need as possible). Table 1 shows the results of using the custom initialization. Length refers to the total length of the trips (in meters), duration (in minutes) refers to the maximum of the trucks' travel time assuming they use a constant speed of 45 km/h and execution is the time taken (in minutes) by the algorithm.

Table 1: Optimization results (average of 10 runs, best values in bold); length is the total distance; duration is the time taken by the truck that takes the longest trip, execution refers to execution time.

Random Generation	Length	Duration	Execution
Hill Climbing	29,374	21.34	25.56
Simulated Annealing	28,914	20.45	24.21
Evolutionary Algorithm	28,645	22.46	24.15
Custom Generation			
Hill Climbing	21,713	16.87	17.89
Simulated Annealing	**21,522**	16.53	18.28
Evolutionary Algorithm	21,606	**16.49**	**16.42**

4 CONCLUSIONS

Inventory rebalancing studies for mobility sharing systems take into account the total distance travelled and try to minimize the number of trucks but usually forget other business constraints (e.g., truck capacities and operational costs). This paper shows an initial attempt to realistically handle this task via metaheuristics.

While interesting results were achieved, our approach contains several limitations and thus further research is required. This study assumes that operational costs are proportional to the number of handled scooters. Yet, the operational costs should be more realistic, involving a fixed cost of parking and a variable one of scooter handling. Also, the optimization system should prioritize restocking stations over removing excess scooters, since it is more important not to lose clients due to the lack of scooters. Moreover, the number of trucks should be minimized, because it reduces expenses with maintenance and personnel for driving and performing the loading operations. There is a trade-off between human resources and operative costs because more trucks mean distribution of the loading tasks. Thus, our next goal will be to modify the fitness function in order to better model this business aspect. As for the metaheuristics, we wish to consider algorithms that operate over graphs and that allow the use of heuristics for the choice of the next station (e.g., ant colony optimization). It is also important to study multi-objective algorithms, since this problem has several conflicting goals (e.g., minimizing the time needed to perform the loading operations, minimizing the number of trucks and total distance travelled).

REFERENCES

[1] Jan Brinkmann, Marlin W. Ulmer, and Dirk C. Mattfeld. 2016. Inventory Routing for Bike Sharing Systems. *Transportation Research Procedia* 19 (2016), 316–327.
[2] Paulo Cortez. 2014. *Modern optimization with R.* Springer.
[3] Mauro Dell'Amico, Eleni Hadjicostantinou, Manuel Iori, and Stefano Novellani. 2014. The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. *Omega* 45 (June 2014), 7–19.
[4] Damianos Gavalas, Charalampos Konstantopoulos, and Grammati Pantziou. 2016. Design and management of vehicle-sharing systems: A survey of algorithmic approaches. In *Smart Cities and Homes*. Elsevier, 261–289.
[5] Lei Shi, Yong Zhang, Weina Rui, and Xinzheng Yang. 2019. Study on the bike-sharing inventory rebalancing and vehicle routing for bike-sharing system. *Transportation research procedia* 39 (2019), 624–633.